M. Tech (VLSI Design & Embedded System) Madhur Gupta 2018

REAL TIME RECOGNITION OF COLORED AIR WRITTEN CHARACTERS

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF

MASTER OF TECHNOLOGY
IN
VLSI Design & Embedded System

Submitted by:

Madhur Gupta

2k16/VLS/13

Under the supervision of

Dr. N. Jayanthi

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

Session 2016-18

DEPARTMENT OF ELECTRONICS &COMMUNICATION ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, Madhur Gupta, Roll No. 2k16/VLS/13 student of M.Tech. (VLSI Design & Embedded System), hereby declare that the project Dissertation titled "Real Time Recognition of Colored Air Written Characters" which is submitted by us to the Department of Electronics and Communication Engineering, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associate ship, Fellowship or other similar title or recognition.

Place: Delhi **(NAME OF STUDENT)**

Date: MADHUR GUPTA

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Project Dissertation titled "Real Time Recognition of Colored Air Written Characters" which is submitted by Madhur Gupta, Roll No 2k16/VLS/13, Electronics & Communication Engineering, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the students under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi

Date:

Dr. N. Jayanthi

(**SUPERVISOR)**

ASSISTANT PROFESSOR

ABSTRACT

Character Recognition has been comprehensively studied in the last half century and has advanced to a level, adequate to produce technology driven applications. Now, the advancement in computational power enables the implementation of the present Character Recognition methodologies and also creates an increasing demand on many emerging application domains, which require more advanced methodologies. Researchers have worked on many methodologies for Recognizing Text or Character from a handwritten or already computer printed piece of paper to store the data obtained in digital form.

In this project our main aim was on recognizing characters in real time from the user's gesture when he is in his writing phase. For implementing the aforesaid work the user's gesture needed to be traced on a real time basis for getting a base to recognize the letter which he is writing. The writing part is realized on a paperless scheme in which the user only need to air write the characters which he wants to write by just affixing any colored piece of object to his finger. Tracing part of letters has been deployed on the colored object which is being attached to the user's finger. The gestures of hand (object attached to it) in real time is captured using camera and is simultaneously being fed to Matlab for further processing.

Our preliminary aim of detecting the colored object and then tracking is performed using the color identifying algorithm, which uses background subtraction strategy, noise filtering, binary image conversion and blob extraction to

recognize a specific color in the video feed. Then the corresponding pixel on the GUI (Graphic user interface) is being draw to track where all the color has been. When a complete character is drawn, a signal is send to Matlab by the user for the next part of recognizing that character. Further work of recognizing the letters is carried out by using the traced part of the object. This traced path on the GUI is sent in the form of an image for further processing to extract the letters written on it. The part of extracting characters from this image is performed using template matching technique. This technique uses a database of characters and numbers, which needs to be stored in their binary from in an M x N cell matrix to be used as templates for comparison purpose. Further the traced image is gone through various preprocessing steps such as Grayscale conversion, Filtering and Feature Extraction which further includes Row detection, Character boundary detection, segmentation and binary conversion. Finally the binary converted character is resized to M x N cell and then correlation method is used for matching this with saved templates. Finally the characters extracted from air written gestures is saved to a text file and also gets displayed simultaneously.

This work might have applications in various domains such as interactive learning sessions in classrooms which no longer will require tutor to be cling to board; effortlessly making notes or writing any information and then saving it only by air writing the gestures in one's own way without even the need of touching the keyboard, Such applications are beneficial for disabled people; transmitting the relevant information from a distance in some form of encoded character or symbols in utter need of secrecy for military purpose and many more.

ACKNOWLEDGMENT

It gives me immense pleasure to express my deepest sense of gratitude and sincere thanks to my highly respected and esteemed guide **Dr. N. Jayanthi**, Asst. Professor, Department of Electronics and Communication Engineering, Delhi Technological University, for giving me all support and guidance which made me complete the project duly. I am extremely thankful for her valuable guidance, encouragement and help for completing this work. Her useful suggestions for this whole work and co-operative behavior are sincerely acknowledged.

I also express my profound sense of appreciation to **Prof. S. Indu** (H.O.D., ECE), for giving encouragement and opportunity to complete my project smoothly.

I am thankful to and fortunate enough to get constant encouragement, support and guidance from all Teaching staffs of ECE Department which helped me in successfully completing my research work. Also, I would like to extend our sincere esteems to all staff in laboratory for their timely support.

Madhur Gupta
2K16/VLS/13
M.Tech (2016-2018)

CONTENTS

LIST OF FIGURES

.

LIST OF TABLES

LIST OF ABBREVATIONS

ANSI: American National Standard Institute

ASCII: American Standard Code for Information Interchange

BMP: Bitmap Image format

CR: Character Recognition

ECMA: European Computer Manufacturers Association

GUI: Graphical User Interface

JPEG: Joint Photographic Experts Group

MATLAB: Matrix Laboratory

OCR: Optical Character Recognition

PNG: Portable Network Graphics

CHAPTER 1

INTRODUCTION

1.1 GENERAL :

Motion gestures provide a complimentary modality for general human computer interaction. Motion gestures are meant to be simple so that a user can easily memorize and perform them. However, motion gestures themselves are not expressive enough to input text for motion-based control. “Air-writing” is defined as writing letters or words with hand or finger movements in a free space. It is especially useful for user interfaces that do not allow the user to type on a keyboard or write on a track-pad/ touchscreen, or for text input for smart system control, among many applications. Isolated letters written in the air involve a sequence of hand movements. Although any snapshot of such movements can be considered a realization of a motion gesture, air-writing is more complicated than hand-gesture recognition because of the interdependence among the involved “hand-gestures” [1].

Tracking of Air writing with the use of color object involves a primary process of tracking that object, so object tracking is an important task in our work. It is considered to be significant within the field of Computer vision. The invention of faster computers, the availability of inexpensive and good-quality video cameras, and the demand for automated video analysis have made object-tracking techniques quite popular. Generally, the video analysis procedure has three major steps:

1) Object detection;
2) Tracking its movement from frame to frame;
3) Analyzing the behavior of the object.

For object tracking, four different issues are taken into account: selection of a suitable object representation, feature selection for tracking, object detection, and object tracking [2]. In the real world, object-tracking algorithms are the primary components of different applications, such as automatic surveillance, video indexing, and vehicle navigation systems [2].

In its simplest form, tracking can be summarized as the problem of estimating the trajectory of an object in the image plane as it moves around a scene. In other words, a tracker assigns consistent labels to the tracked objects in different frames of a video [2]. In addition, depending on the tracking domain, a tracker can also provide object-centric nformation, such as orientation, area, or shape of an object. Tracking objects can be complex due to some ssues regarding; noise an mage, complex motion of object, complex shape of object, real-time processing requirements etc.

One can simplify tracking by imposing constraints on the motion and/or appearance of objects [2]. For our work of Character recognition in real time as we cannot put constraint on the object motion, we have restrained ourselves on the appearance of object through which gestures are being made by limiting its choice of color. So, one approach to this problem is to convert the retrieved RGB frame into corresponding HSV (Hue-Saturation-Value) plane and then extract the pixel values only for the some specific decided color like RED. Then choose a range which covers all the different shades of red. In this way detection of almost all distinguishable color of object through which one want to Air write is possible, so selection of any colored object which is readily available can be chosen.

1.2 COLOR MODEL

A digital image that includes a specific intensity for each individual pixel and that too in three or four columns is known as a color image. For such results which can be deciphered easily and are acceptable visually and also sufficient to give us with at least three samples or three intensities for each different color space which can be thought of to be coordinates of the color space. The most commonly used display is the RGB color display and it is even used in computers, laptops etc. Other color spaces however are also used such as the CMYK and the HSV and they can be used in other contexts such as newspapers etc. An image when defined in terms of color will have three intensities each for a different color and can even measure the light's chrominance [3].

Whenever a model of color adds to the previous intensity it is known as an additive model of color. In RGB all colors are added together to form a large variety of colors and provide us with the amazing visual we often see.

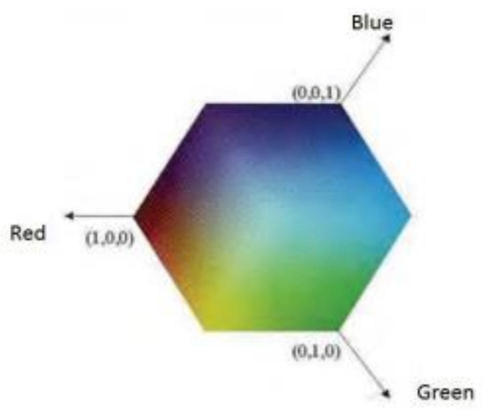


Fig. 1.1 RGB Color space where x-axis represent red, y-axis is green and z-axis is blue. [3]

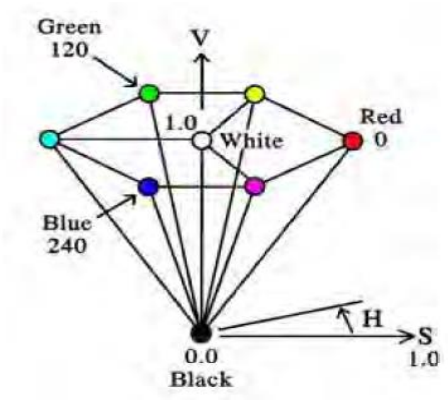


Fig. 1.2 HSV color Space : Value(V) is represented by axis orthogonal of the plane, Angle represents Hue, radius represents level of Saturation (purity) [3].

RGB comes from the first letters of all the three colors it contains i.e. Red, Green and Blue. This color model uses light to create color, and it is used for digital media. When you play a game on your smart phone or watch an action movie on TV, you are seeing color in an RGB color space. This model will always depend on the device it is running and it may be interpreted or reproduced in different manners because of the different color elements behind their usage. Even the single levels of each color may differ from producer to producer. Thus the same model can never guarantee the same display reproduction across devices.

1.3 CRARACTER RECOGNITION

Optical character recognition (OCR) is a process of converting a printed document or scanned page into ASCII characters that a computer can recognise. Computer systems equipped with such an OCR system improve the speed of input operation, decrease some possible human errors and enable compact storage, fast retrieval and other file manipulations. Its range of application includes postal code recognition, automatic data entry into large administrative systems, banking, automatic cartography and reading devices for blind.

Accuracy, flexibility and speed are the main features that characterise a good CR system. Several algorithms for character recognition have been developed based on feature selection**.** The performance of systems has been constrained by the dependence on font, size and orientation. The recognition rate in these algorithms depends on the choice of features. Most of the existing algorithms involve extensive processing on the image before the features are extracted that results in increased computational time [5]. In our dissertation a pattern matching based method has been deployed for character recognition that would effectively decrease the image processing time while maintaining efficiency and versatility. The key factors involved in the implementation are: an optimal selection of features which categorically defines the details of the characters, the number of features and allow image processing time.

1.4 CR System Classification:

In reference [6] Character Recognition systems have been classified according to the data acquisition techniques and the text type as follows:

1.4.1 Systems Classified According to Data Acquisition"Techniques :

The progress in CR methodologies evolved in two categories according to the mode of data acquisition, as online and off-line character recognition systems.

a) Online Character Recognition Systems:

The problem of recognizing handwriting, recorded with a digitizer, as a time sequence of pen coordinates is known as "on-line character recognition". The digitizers are mostly electromagnetic-electrostatic tablets, which send the coordinates of the pen tip to the host computer at regular intervals. There are also, other technologies including laser beams and optical sensing of a light pen. The on-line handwriting recognition problem has a number of distinguishing features, which must be exploited to get more accurate results than the off-line recognition problem as discussed in [6]:

- Real time process.
- Adaptive nature in real time.
- Captures dynamic information of trajectory of pen.
- Pre-processing requirement is very less.

These systems are useful in social environments where speech does not provide enough privacy. They provide an efficient alternative for the large alphabets where the keyboard is cumbersome. Pen based computers [7], educational software for teaching handwriting [8] and signature verifiers [9] are the examples of popular tools utilizing the on-line character recognition techniques.

b) Off - Line Character Recognition Systems:

Off-line character recognition is known as Optical Character Recognition (OCR), because the image of writing is converted into bit pattern by an optically digitizing device such as optical scanner or camera [6]. The recognition is done on this bit pattern data for machine - printed or hand-written text. The research and development is well progressed for the recognition of the machine-printed documents. In recent years, the focus of attention is shifted towards the recognition of hand-written script.

The drawbacks of off-line character recognition as compared to online recognition are summarized as follows:

- It requires imperfect and costly pre-processing techniques prior to feature and recognition stages.
- The lack of progressive or dynamic information results in lower recognition rates as compared to online recognition.

Some applications of the off-line recognition are large-scale data processing such as postal address reading [10], check sorting [11], Office automation for text entry [12], automatic inspection and identification [13], etc. It is a very important tool for creation of the electronic libraries. It provides a great compression and efficiency by converting the document image from any image file format into more useful formats like HTML or various word processor formats.

1.4.2 **Systems Classified According to Text Type:**

Hand-written and machine-printed are the two main areas of interest in CR filed considering text type character recognition systems.

Machine-printed text includes the materials such as books, newspapers, magazines, documents and various writing units in the video or still image. The problems for fixed-font, multi-font and Omni-font character recognition is relatively well understood and solved with little constraint [14], [15], [16]. When the documents are generated on a high quality paper with modern printing technologies, the available systems yield as good as 99% recognition accuracy. However, the recognition rates of the commercially available products are very much dependent on the age of the documents, quality of paper and ink, which may result in significant data acquisition noise.

On the other hand, hand-written character recognition systems have still limited capabilities. The problem can be divided into two categories: cursive and hand-printed script [6]. In practice, however, it is difficult to draw a clear distinction between them. A combination of these two forms can be seen frequently. Based on the nature of writing and the difficulty of segmentation process, Tappert [17] has defined five stages for the problem of handwritten word recognition as indicated in Fig. 1.3.

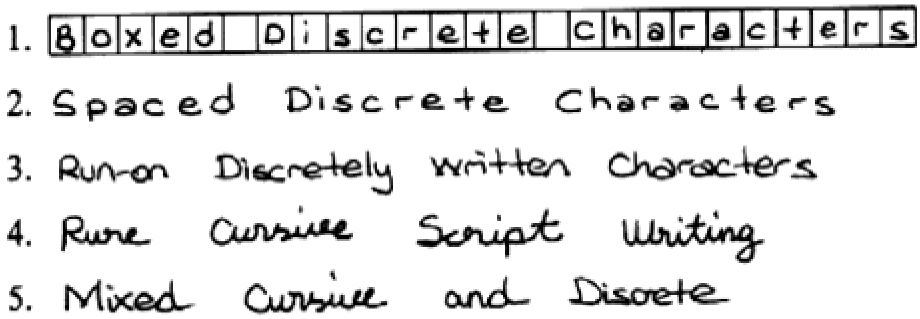


Fig. 1.3 Five stages of handwritten word recognition problem.

Boxed discrete characters require the writer to place each character within its own box on a form. The boxes themselves can be easily found and dropped out of the mage or can be printed on the form in a special color ink that will not be picked up during scanning, thus eliminating the segmentation problem entirely. Spaced discrete characters can be segmented reliably by means of horizontal projections, creating a histogram of gray values in the image over all the columns and picking the valleys of this histogram as the points of segmentation. This has the same level of segmentation difficulty as is usually found with clean machine- printed documents. Characters at the third stage are usually discretely written, however they may be touching, therefore making the points of segmentation less obvious. Cursively or mixed written texts require more sophisticated approaches compared to the previous cases. First of all, advanced segmentation techniques are to be used for character based recognition schemes. In pure cursive handwriting, a word is formed mostly from a single stroke. This makes segmentation by the traditional projection or connected component methods ineffective. Secondly, shape discrimination between character that look alike, such as U- V, I-1, O-0, is also difficult and requires the context information.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

Writing, which has been the most natural mode of collecting, storing and transmitting the information through the centuries, now serves not only for the communication among humans, but also, serves for the communication of humans and machines. The intensive research effort on the field of CR has been carried on from a last some decades.

` Character recognition is not a new problem but its roots can be traced back to systems before the inventions of computers. The earliest OCR systems were not computers but mechanical devices that were able to recognize characters, but very slow speed and low accuracy. Reference [18] describes, In 1951, M. Sheppard invented a reading robot GISMO that can be considered as the earliest work on modern OCR. It can read musical notations as well as write words on a printed page one by one. However, it can only recognize 23 characters. The machine also has the capability to could copy a typewritten page. In 1954, J. Rainbow devised a machine that can read uppercase typewritten English characters, one per minute. The early OCR systems were criticized due to errors and slow recognition speed. Hence, not much research efforts were put on the topic during 60's and 70's.

Because of the complexities associated with recognition, it was felt that three should be standardized OCR fonts for easing the task of recognition for OCR. Hence, OCRA and OCRB were developed by ANSI and ECMA in 1970 that provided comparatively acceptable recognition rates [19]. American standard OCR character set was defined as OCR–A shown in Fig. 2.1. This font was highly stylized and designed to facilitate optical recognition although still readable to humans. A European font design was called as OCR–B as shown in Fig. 2.2 which had more natural fonts than American standard. Attempts were made to merge two fonts in one standard through machines which could read both standards.

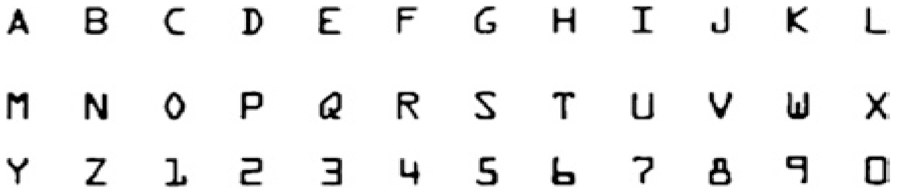


Fig. 2.1 Font of OCR-A. [19]

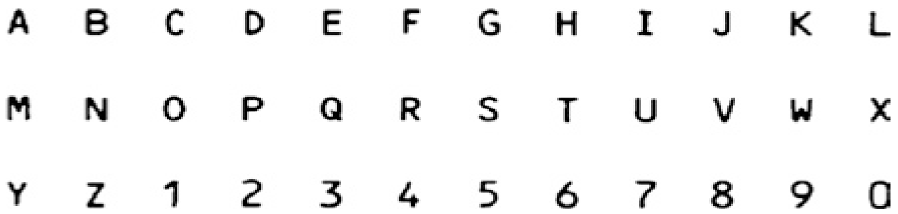


Fig. 2.2 Font of OCR-B. [19]

During the past thirty years, substantial research has been done on OCR. This has led to the emergence of document image analysis (DIA), multi-lingual, handwritten and omni-font OCRs [20]. Despite these extensive research efforts, the machine's ability to reliably read text is still far below the human. Hence, current OCR research is being done on improving accuracy and speed of OCR for diverse style documents printed/ written in unconstrained environment.

2.2 LITERATURE REVIEW

- **Hannuksela** *et al.* [21], presented finger-tracking-based character recognition systems, respectively. The author presents a motion-based tracking algorithm that combines the Kalman filtering and expectation–maximization techniques for estimating two distinct motions: those of the finger and the camera. The estimation is based on the motion features, which are effectively computed from the scene for each image frame. Their main idea is to control a mobile device simply by moving a finger in front of a camera.

- **T. Asano** *et al.* [22], discuss a visual interface that recognizes the Japanese katakana characters in the air. For tracking the hand gesture, they used an LED pen and a TV camera. They converted the movements of the pen into direction codes. The codes were normalized to 100 data items to eliminate the effect of the writing speed, in which they defined 46 Japanese characters.

- **S. Vikram** *et al.* [23], presented a new recognition system for gesture and character input in the air. For detecting the finger positions, a 3D capturing device called leap motion controller is used. In the method proposed in such paper, the dynamic time warping (DTW) distance technique is used for searching a similar written character from the database. For character recognition, a database consisting of 29,000 recordings was created, in which simple pre-written characters are present.

- **Faisal Mohammad** et.al. [24], presented pattern matching algorithm for typewritten and handwritten characters. The binary image is divided into 5 tracks and 8 sectors. The track-sector matrix is then matched with existing template. The existing template consists of each track-sector intersection value, each track value and each sector value. If all these parameters are found to match with the template values then the resultant is the character identified.

- **Faisal Baig** et.al. [25], presented Text writing in air through the mobile camera recognizing English handwritten characters by hand gestures. First, it tracks the colored fingertip in the video frames and then applies English optical character reorganization over the plotted images to recognize the written characters using Template matching. The proposed algorithm showed an average accuracy rate of 92.083% when tested for different shaped alphabets on 3000 dataset of characters.

- **M. Ziaratban** et.al. [26], proposed an approach for character recognition termed as template matching. This technique extracts feature by searching the special templates in input images. For each template, the amount of matching is used as feature and position of the best matching in an image is found and saved.

CHAPTER 3

DETECTION AND TRACKING

3.1 INTRODUCTION

It is the process of localizing an object or multiple objects using a single camera, multiple cameras or from a video file already at hand and then tracking it depending on various applications depending on the user's need. Technically, in object tracking we estimate or define the trajectory or path of an interested object in the frame plane as it moving around the image plane. Because of technology increasing in computational power, availability of good quality and low cost video camera and the need of automated video system people are sowing the more interest in object tracking algorithm. In a video analysis, there are three basic steps:

1. Recognition of intended object from moving objects,
2. Tracking of our interested objects in consecutive frames, and
3. Analysis of trajectory to for further processing or understanding the nature of interested object.

This process is a preliminary part of our work in which any object readily available is used a pen for air writing characters. So our primary task is to detect that object at hand and track it for further process of detecting the characters and to write them in a Text file. This section will help to develop our knowledge in various methods for detection of object and it's tracking techniques.

3.2 CHALLENGES IN DETECTION AND TRACKING

Properly detecting objects can be a particularly challenging task, especially since objects can have rather complicated structures and may change in shape, size, location and orientation over subsequent video frames. Various algorithms and schemes have been introduced in the few has their own advantages and drawbacks. Any object tracking algorithm will contain errors which will eventually cause a drift from the object of interest. The better algorithms should be able to minimize this drift such that the tracker is accurate over the time frame of the application.

In object tracking the important challenge that has to consider while the operating a video tracker are when the background is appear which is similar to interested object or another object which are present in the scene. This phenomenon is known as clutter. The other challenges except from cluttering may difficulty to detect interested object by the appearance of that object itself in the frame plane due to factors which are described as follow:

- Appearance of object in video frame,
- Ambient illumination of object in video frame,
- Noise level of image or video signal,
- Blockage of moving object behind other object.

3.3 REPRESENTATION OF OBJECT

Object tracking is a video processing application with a wide number of applications. Applications may include tracking particular people in a video for security reasons for tracking planetary objects from satellite data for astronomical studies. An object of interest is defined on the basis of particular application which is present at hand. An object of interest may depend on the type of application. For example, in our work of real time recognition of air written characters our object of interest is anything which the user holds to write characters in air, whereas for surveillance of traffic application our object would be human or car or for gaming application it could be the face of some particular person.

In an object tracking algorithm, an object of interest is defined on the basis of application which are present but it can be used for further analysis. From previous example, it is clear that we have to take such object as objects of interest which help to object tracking. An interested object may be modelled by their appearance and shape. In object representation interested object can be modelled different way that can help to video tracker [27], some of those ways are listed below:

- **Point:** Interested object can be represented by using a point. In this we can represent our interested object by either single point (e.g. centroid) (Fig. 3.1(a)) or multiple points (Fig.3.1(b)). We can use point representation of interested object in those tracking object application where target is present in smaller region or target is itself small.
- **Primitive geometric shapes:** Interested object can be represented by geometric shape. For example, by using circle, ellipse, rectangle etc.(Fig.2(c),(d)). Primitive geometric shape representation can be used for representing a rigid object; it can also useful for tracking simple non rigid object. These types of representation are modelled by projective transformation, affine or translation for object motion.
- **Object silhouette and contour:** Interested object can be represented by contour (Fig.2 (g), (h)). The boundary of the object is defined as contour representation (Fig. 2(i)). Region surrounded by the contour is known as the silhouette of interested object. To represent the complex non-rigid shapes we generally use contour or silhouette representation.
- **Templates:** Template model can be created by using the basic shape of geometry or object silhouettes. This is a better representation of an object because it can contain both appearance and special information about interested object.

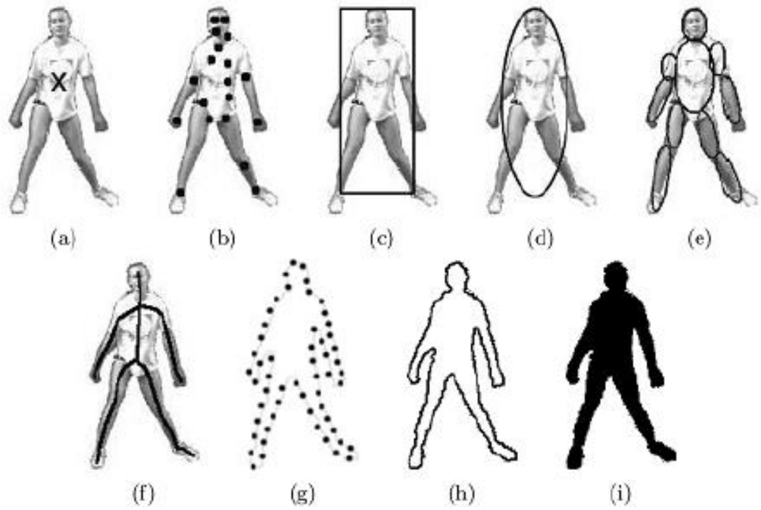


Fig. 3.1 Different representation of an object. [27]

However, there is a solid relationship between the interested object representations and the tracking algorithm. Object representations are generally picked as indicated by the requisition space. For tracking interested object which seem a little in a frame, point representation is normally fitting. For the object whose shapes could be approximated by rectangles or circles, primitive geometric shape representations are more suitable.

3.4 DETECTION OF OBJECT

Every tracking method requires an object detection mechanism either in every frame or when the object first appears in the video. A common approach for object detection is to use information in a single frame. However, some object detection methods make use of the temporal information computed from a sequence of frames to reduce the number of false detections. For object detection, there are several common object detection methods described in [28].

1. **Point detectors**: Point detectors are used to find interesting points in images which have an expressive texture in their respective localities. A desirable quality of an interest point is its invariance to changes in

illumination and camera viewpoint. In literature, commonly used interest point detectors include Moravec's detector, Harris detector, KLT detector, SIFT detector.

2. **Background Subtraction**: Object detection can be achieved by building a representation of the scene called the background model and then finding deviations from the model for each incoming frame. Any significant change in an image region from the background model signifies a moving object. The pixels constituting the regions undergoing change are marked for further processing. This process is referred to as the background subtraction.
3. **Segmentation:** The aim of image segmentation algorithms is to partition the image into perceptually similar regions. Every segmentation algorithm addresses two problems, the criteria for a good partition and the method for achieving efficient partitioning. In the literature survey it has been discussed various segmentation techniques that are relevant to object tracking [27]. They are, mean shift clustering, and image segmentation using Graph-Cuts (Normalized cuts) and Active contours.

Object detection can be performed by learning different object views automatically from a set of examples by means of supervised learning mechanism.

3.5 TRACKING OF OBJECT

The aim of an object tracker is to generate the trajectory of an object over time by locating its position in every frame of the video [27]. The tasks of detecting the object and establishing a correspondence between the object instances across frames can either be performed separately or jointly. In the first case, possible object region in every frame is obtained by means of an object detection algorithm, and then the tracker corresponds objects across frames. In the latter case, the object region and correspondence is jointly estimated by iteratively updating object location and region information obtained from previous frames [27].

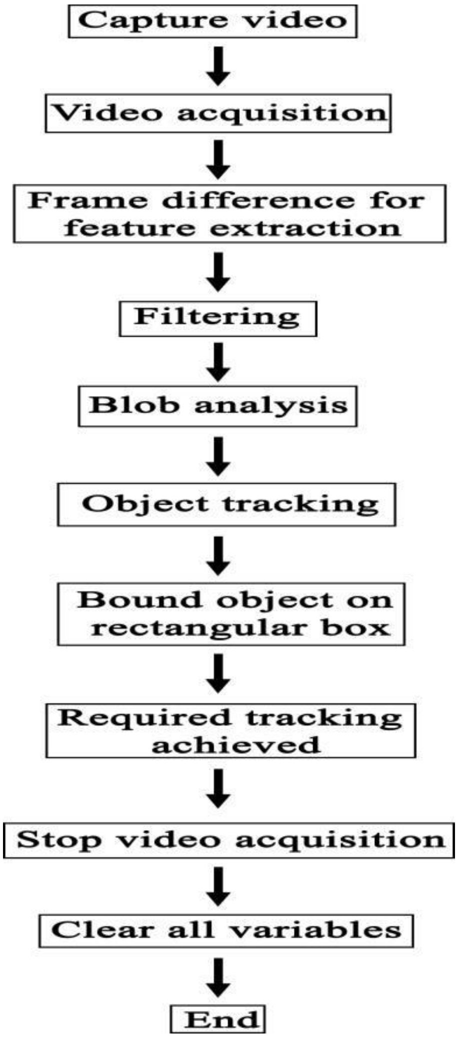


Fig. 3.2 Flowchart for object detection and tracking using color feature. [28]

Feature selection plays a vital role to select a. proper feature in tracking. So feature selection is closely related to the object representation. For example, color is used as a feature for histogram based appearance representations, while for contour-based representation object edges is usually used as features. Generally, many tracking .algorithms uses .a combination .of these features. Some common visual features are detailed as following:

- **Color**: Color of an object is influenced by two factors. They are Spectral power distribution of the illuminant and Surface reflectance properties of the object. Different color models are RGB, L*u*v and L*a*b used to represent color.

- **Edges**: Edge detection is used to identify strong changes in image intensities generated by object boundary. Edges are less sensitive to illumination changes compared to color features. Most popular edge detection approach is Canny Edge detector.
- **Optical Flow**: It is a dense field of displacement vector which defines the translation of each pixel in a region. It is computed using the brightness constraint, which assumes brightness constancy of corresponding pixels in consecutive frames. Optical Flow is commonly used as a feature in motion based segmentation and tracking application.
- **Texture**: Texture is a measure of the intensity variation of a surface which quantifies properties such as smoothness and regularity. It requires a processing step to generate the descriptors. There are various texture descriptors: Gray-Level Co-occurrence Matrices, loss texture measures, wavelets, and steerable pyramids.

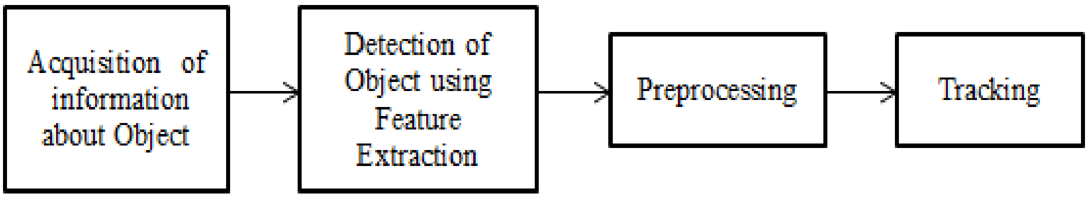


Fig. 3.3 Block diagram for Object Tracking. [29]

Mostly features are chosen manually by the user depending on the application. In our work we have chosen Color as tracking feature which enable user to take any suitable thing regardless of its structure and use that as pen for air writing. After tracking of the selected object gets completed, it is simultaneously saved in any image format such as .JPEG, .BMP, etc. for further processing. The basics of extracting characters or words using image processing after tracking is being accomplished is explained in the next chapter.

3.6 ALGORITHM IMPLEMENTED

3.6.1 Capturing Image From Live Video Feed

- First we initiate the video stream in our MATLAB project and give parameters to it to return to us an RGB image.
- We take continuous snapshots (images) from the video stream and process them each in the same manner.
- This is taken care of with the command ‘step’.
- Each matrix returned by using the above command is of the size of the resolution of the video stream with each R, G and B component. Size of the matrix returned is: [480 640 3].

3.6.2 Extraction of Red Component

- Get only the red component of the image by subtracting the grey scale values from the red column of the matrix.
- Grayscale image is a type of an image in which the value of each pixel will be a single value unlike the RGB model, such sort of images can be said to be a mixture of black and white and their different combinations thus representing many shades of grey. This shade of grey will be black at the lowest intensity and white at the highest.
- This type of grayscale images are very different form normal black and white images if we consider the view point of a computer and its imaging. Black and white images are known as bi-level since they have only two values whereas a grey image has lots of shades.

3.6.3 Filtering Noise

- Any random variation in any property of the image from the general pattern or the surrounding region is known as Noise. This can be developed as a result of any electronic disturbance. Any sensor or wrong circuit can trigger this in any device.

- Although we could have used any other filters such as the Gaussian filter or the mean filters but they are not able to enhance the edges and remove the noise uniformly. For this purpose we have used the median filter which can produces edges of higher quality as compared to the other filters.

- **<u>Median Filter</u> :**

  - It is one of the most used filters because it is better at removing noise than the other filters available. Also after removal the edge quality is better in this filter.
  - Now let me explain the working and the idea behind a median filter. What it does is it puts all the values surrounding the pixel into an array, if it is a boundary pixel it appends that many zeroes to the array, now it arranges the array in increasing order and then selects the median of the array. The original pixels whose neighbors were taken out are now replaced with a new value which is the value of the median, the array containing all the neighbors is known as the "window". Process is iterated again for each one.
  - Working of a median Filter has been shown below:

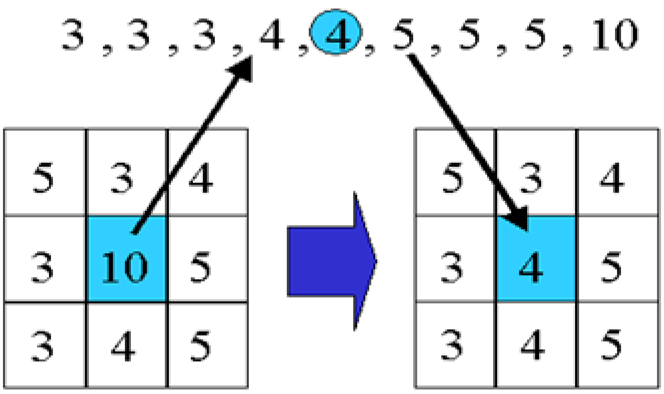


Fig.3.4 Working of Median filter

  - The same process is repeated for each pixel of the image.
  - For pixels on the boundary the empty pixels are given a value of zero.

3.6.4 **Converting Normal Image to Binary Image**

- The image is then converted to a binary image by using a particular threshold of the Red level.
- This conversion to binary is done to make the further steps less complex to solve.
- The threshold for the Red color is decided by running it at multiple thresholds and then deciding the optimum one based on the lighting of the room.

3.6.5 **Blob Analysis**

- The methods of detecting blobs are working using a very simple principle. What it does is that it detects regions in the image which are very different in properties such as brightness or any other thing in comparison to the region that surrounds it.
- Basic Scenario of blob analysis consists of three major steps:
    - Extracting the region.
    - Refining the blob identified.
    - Analyzing the final blob.
- We are only left with now the 3rd step.

3.6.6 **Recognition**

- Once the blob is recognized the function returns the bounding box and the centroid of the blob.
- We mark both the centroid and the blob on the video stream.
- We have also defined a threshold for the maximum and the minimum area of the blob.

CHAPTER 4

CHARACTER RECOGNITION

4.1 INTRODUCTION

In the running world, there is growing demand for the software systems to recognize characters in computer system when information is scanned through paper documents as we know that we have number of newspapers and books which are in printed format related to different subjects. These days there is a huge demand in storing the Information available in these paper documents in to a computer storage disk and then later reusing this information by searching process. One simple way to store information in these paper documents in to computer system is to first scan the documents and then store them as Images. But to reuse this information it is very difficult to read the individual contents and searching the contents form these documents line-by-line and word-by-word. As a result, computer is unable to recognize the characters while reading them. This concept of storing the contents of paper documents in computer storage place and then reading and searching the content is called Document Processing. For this document processing there are various techniques in the world. Amongst all those techniques we have chosen Optical Character Recognition as main fundamental technique to recognize characters.

Optical character recognition (OCR) is the recognition of printed or handwritten text by a computer. This involves photo scanning of the text character-by-character, analysis of the scanned-in image, and then translation of the character image into character codes, such as ASCII, commonly used in data processing. In OCR processing, the scanned-in image or bitmap is analyzed for light and dark areas in order to identify each alphabetic letter or numeric digit. When a character is recognized, it is converted into an ASCII code. Optical character recognition is needed when the information should be readable both to humans and to a machine and alternative inputs cannot be predefined. In comparison with the other techniques for automatic identification, optical character recognition is unique in that it does not require control of the process that produces the information.

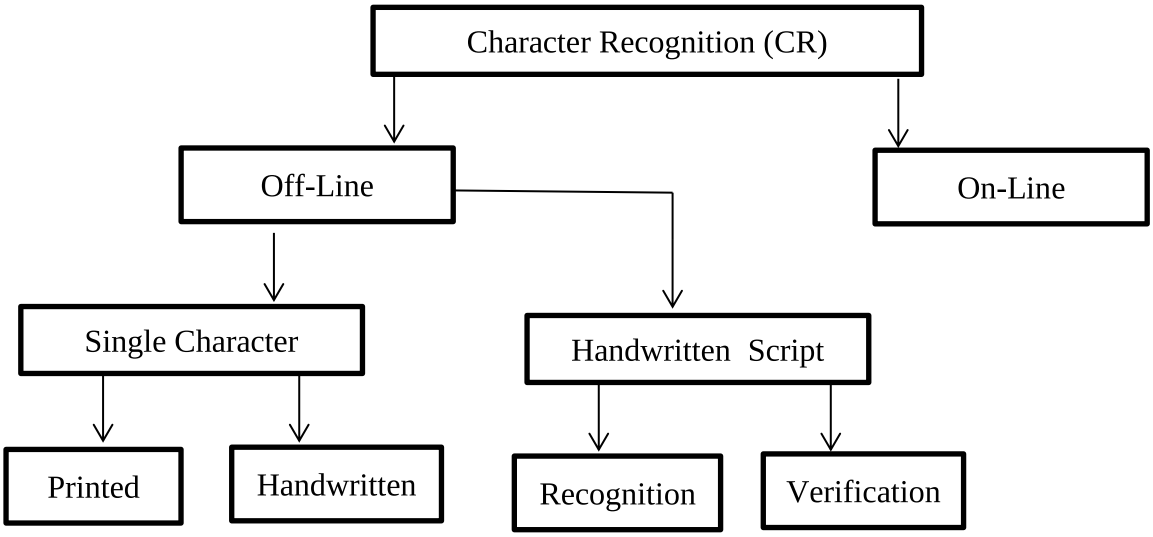


Fig. 4.1 Different areas of character recognition. [30]

Optical Character Recognition deals with the problem of recognizing optically processed characters. Optical recognition is performed off-line after the writing or printing has been completed, as opposed to on-line recognition where the computer recognizes the characters as they are drawn. Both hand printed and printed characters may be recognized, but the performance is directly dependent upon the quality of the input documents.

4.2 CR METHODOLOGY

Character Recognition is the process whereby typed or printed pages can be scanned into computer systems, and their contents recognized and converted into machine - readable code. Template matching is one of the Optical Character Recognition techniques. Template matching is the process of finding the location of a sub image called a template inside an image. Once a number of corresponding templates is found their centres are used as corresponding points to determine the registration parameters. Template matching involves determining similarities between a given template and windows of the same size in an image and identifying the window that produces the highest similarity measure. It works by comparing derived image features of the image and the template for each possible displacement of the template.

CR by using Template Matching is a system prototype that is useful to recognize the character or alphabet by comparing two images of the alphabet. The objectives of this system prototype are to develop a prototype for the Optical Character Recognition (OCR) system and to implement the Template matching algorithm in developing the system prototype. This system prototype has its own scopes which are using Template Matching as the algorithm that applied to recognize the characters and numerals, characters to be tested are Uppercase and Lowercase English alphabets (‘A - Z’, ‘a -z’). Grey-scale images are being used with Times New Roman font type, using bitmap image format with 42 x 24 image size and recognizing the alphabet by comparing between the extracted image and these images. The purpose of this system prototype is to solve the problem in recognizing the character which is before that it is difficult to recognize the character without using any techniques and Template Matching is as one of the solution to overcome the problem.

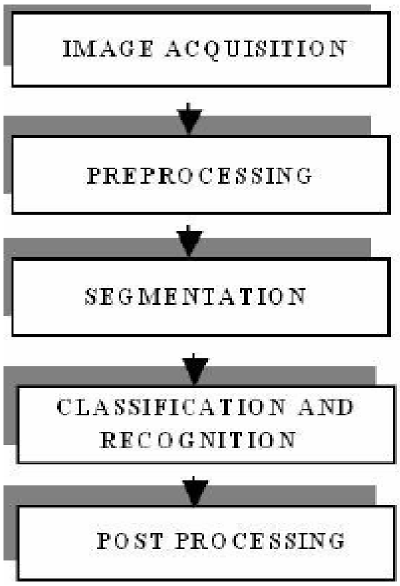


Fig 4.2 Steps of Recognition system without Feature extraction.[31]

There are a few processes that are involved in this system prototype [32] which can be seen in Fig. 4.2 and are explained as follows:

4.2.1 Image Acquisition/ Capture Image

In Image acquisition, the recognition system acquires a scanned image as an input image. This first step is the capturing of an image using electronic devices such as optical (digital/video) camera; webcam etc. can be used to capture the acquired images. For this work, air written characters which are being traced using a Laptop or PC's camera is saved on a Figure window. This traced frame can be saved as an image in different Image formats such as colour JPEG, BMP or PNG. Next, we might proceed in using the Matlab function to convert the traced JPEG image into gray scale format. Input of this system is the image captured by a camera integrated in a laptop.

4.2.2 Pre-Processing

After document scanning, a sequence of data pre-processing operations are normally applied to the images of the documents in order to put them in a suitable format ready for feature extraction. Conventional pre-processing steps include noise removal/smoothing, document skew detection/correction, connected component analysis, normalization, slant detection/correction, thinning, and contour analysis. The purpose of the first important step, image, and data pre-processing, of a character recognition system is to prepare sub-images with optimal quality so that further steps can work correctly and efficiently. The raw data depending on the data acquisition type is subjected to a number of preliminary processing steps to make it usable in the descriptive stages of character analysis. The image resulting from scanning process may contain certain amount of noise. As the first important step, image and data pre-processing serve the purpose of extracting regions of interest, enhancing and cleaning up the images, so that they can be directly and efficiently processed by the feature extraction component. The pre- processing component thus aims to produce data that are easy for the OCR systems to operate accurately. It is an important activity to be performed before the actual data analysis.

The main processes of pre-processing can be pointed as follows:

- **Grayscale Conversion:** In this phase of pre-processing, it involves conversion of the colored input image of handwritten characters into a gray image. The method is based on different color transform. According to the R, G, B value in the image, it calculates the value of gray value, and obtains the gray image at the same time. It is converted to grayscale format by using "rgb2gray" function of MATLAB.
- **Binarization:** Binarization is an important image processing step in which the pixel values are separated into two groups; white as background and black as foreground. Only two colors, white and black, can be present in a binary image. The goal of binarization is to minimize the unwanted information present in the image while protecting the useful information. It must preserve the maximum useful information and details present in the image, and on the other hand, it must eliminate the background noise associated with the image in an efficient way. It is assumed that the intensity of the text is less than that of background i.e. the input image has black foreground pixels and white background pixels. The colors can be inverted if the input image has text intensity more than that of background. Also, the background intensity remains almost uniform throughout the whole image and does not change drastically anywhere in the input image. Hence, in the proposed binarization technique, global gray scale intensity thresholding is employed and the resulting handwritten character image is free from any background noise.
- **Compression**: In order to reduce storage requirements and to increase processing speed it is often desirable to represent gray scale or color images as binary images known as 'Thresholding', by picking a threshold value. The two important categories of thresholding are viz global and local. The global thresholding picks one threshold value for the entire character image which is often based on an estimation of the background level from the intensity histogram of the image [6, 32]. The local or adaptive thresholding use different values for each pixel according to the local area

information [6, 32]. A comparison of common global and local thresholding techniques is given by using an evaluation criterion that is goal directed keeping in view of the desired accuracy of the OCR system [6, 32].

4.2.3 Segmentation

The pre-processing stage yields a clean character image in the sense that a sufficient amount of shape information, high compression, and low noise on a normalized image is obtained. The next OCR component is segmentation. Here the character image is segmented into its subcomponents. Segmentation is important because the extent one can reach in separation of the various lines in the characters directly affects the recognition rate. In this stage, an image of sequence of characters is decomposed into sub-images of individual character. In the proposed system, the pre-processed input image is segmented into isolated characters by assigning a number to each character using a labelling process. This labelling provides information about number of characters in the image. Each individual character is uniformly resized into ‘42 X 24’ pixels for classification and recognition stage.

Internal segmentation is used here which isolates lines and curves in the cursively written characters. Though several remarkable methods have developed in the past and a variety of techniques have emerged, the segmentation of cursive characters is an unsolved problem. The character segmentation strategies are divided into three categories [6, 32]: (a) explicit segmentation (b) implicit segmentation and (c) mixed strategies.

a. **Explicit Segmentation**: The segments are identified based on character like properties. The process of cutting up the character image into meaningful components is achieved through dissection. Dissection analyses the character image without using a specific class of shape information. The available methods based on the dissection of the character image use white space and pitch, vertical projection analysis, connected component analysis and landmarks.

b. **Implicit Segmentation**: This strategy is based on recognition. It searches the image for components that matches the predefined classes. The segmentation is performed by using the recognition confidence including syntactic or semantic correctness of the overall result.

c. **Mixed Strategy:** It combine explicit and implicit segmentation in a hybrid way. A dissection algorithm is applied to the character image, but the intent is to over segment i.e. to cut the image in sufficiently many places such that the correct segmentation boundaries are included among the cuts made. Once this is assured, the optimal segmentation is sought by evaluation of subsets of the cuts made.

4.2.4 Classification & Recognition

This is also an important and final step in our work of recognizing air written characters, using a prepared database of lower and upper case characters ('A – Z' & 'a-z') and numerals from '0 - 9'. Apart from the database of characters this step also requires some additional process of Template matching using correlation technique. Some processes involved in this step are explained below:

- **Creation of Database:** Initially a database is created for twenty-six English alphabets in upper and lower cases i.e. total 52 characters are saved in standard font. Similarly we create a database for 10 digits of the standard font. These are alphabets/digits that we consider to be ideal ones. The alphabets/digits are stored in order starting from the numbers and then alphabets of same font. There is no limit to the fonts that can be stored but there exists a trade-off between the size of database and accuracy. Greater is the number of fonts higher the efficiency of the system. The character images are stored in binary format i.e. in black and white form. The character images can also be scanned by using a scanner.

  All the alphanumeric character images are stored to a uniform image format such as .bmp or .jpeg, so as to make all the images ready for the next step. Pure black background may be used to write/print these character images. We can also have characters to be written with different pens of various colored ink.

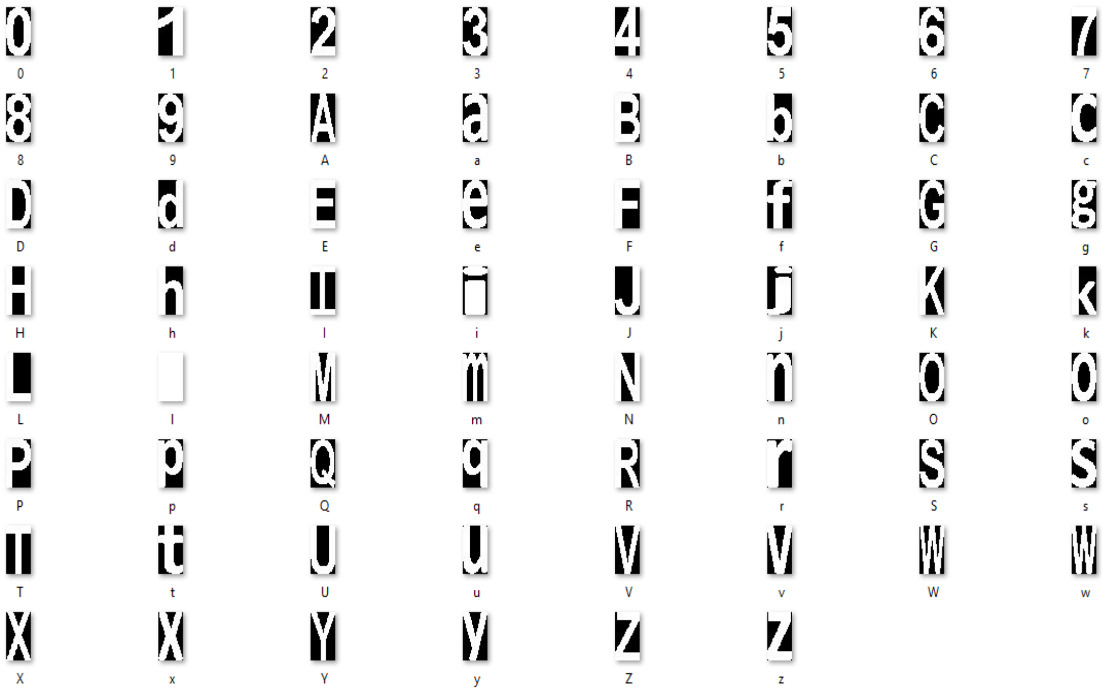


Fig 4.3 Database created for template matching technique.

Character image samples can be contributed by different people of different age groups which could further help in enhancing our recognition of handwritten letters.

- **Template Matching:** One of the important OCR components is training and recognition. OCR systems extensively use the methodologies of pattern recognition which assigns an unknown sample into a predefined class. The OCR is investigated in four general approaches of pattern recognition as suggested in [6, 32]: (a) template matching (b) statistical techniques (c) structural techniques and (d) ANNs. These approaches are neither necessarily independent nor disjointed from each other. Occasionally, an OCR technique in one approach can also be considered to be a member of other approaches. In all of the above approaches, OCR techniques use either holistic or analytic strategies for the training and recognition stages. The holistic strategy employs top down approaches for recognizing the full character eliminating the segmentation problem. The price for this computational saving is to constrain the problem of OCR to limited vocabulary. However, with the cooperation of segmentation stage, the problem is reduced to the recognition of simple isolated characters or strokes, which can be handled for unlimited vocabulary with high recognition rates.

Template matching is a system prototype that useful to recognize the character or alphabet by comparing two images. It is the process of finding the location of sub image called a template inside an image. Template matching involves determining similarities between a given template and windows of the same size in an image and identifying the window that produces the highest similarity measure. The simplest way of OCR is based on matching the stored prototypes against the character to be recognized. Template matching involves the use of a database of characters or templates. There exists a template for all possible input characters. We chose template matching technique because of its less computational complexity.

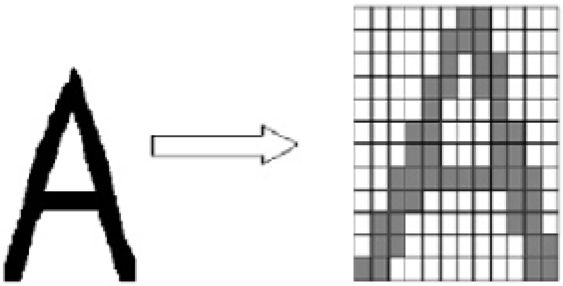


Fig. 4.4 Bitmap image of Character ‘A’.

For recognition to occur, the current input character is compared to each template to find either an exact match or the template with the closest representation of the input character. Template matching algorithm is also known as matrix matching or pattern matching algorithm.

Correlation is a measure of the degree to which two variables or images agree, the two variables are the corresponding pixel values in two images, template and test image [33]. The correlation value will be evaluated by the Equation (4.1) given below:

$$\text{cor}= \frac{\sum_{i=0}^{n-1}(x_i - x).(t_i - t)}{\sqrt{\sum_{i=0}^{n-1}(x_i - x)^2.\sum_{i=0}^{n-1}(t_i - t)^2}} \tag{4.1}$$

In the above equation x_i is the template image and t_i represents test image under consideration and *n* means number of class in the template, number of digits stored in the knowledge base i.e. 62 as shown in Fig. 4.3. The characters of air written text is classified by using similarity index evaluated using correlation. The matching with the knowledgebase x_i and test character t_i image will be done using the Equation (4.1). The value of *y* which carries maximum value with index to represent the class for which the given query character digit image belongs will be evaluated using Equation (4.2).

$$Y = \max \int_{i=1}^{c_n} (cor, i) \tag{4.2}$$

Where c_n denotes the number of different supervised classes under consideration in the proposed methodology its value is 10. [33]

4.2.5 Post-Processing

The last OCR component is post-processing. Some of the commonly used post-processing activities include grouping and error detection and correction. In grouping symbols in text are associated with strings. The result of plain symbol recognition in text is a set of individual symbols. However, these symbols do not usually contain enough information when they are extracted from a random file or a document. These individual symbols are associated with each other making up words and numbers. The grouping of symbols into strings is based on symbols' location in document. The symbols which are sufficiently close are grouped together. For fonts with fixed pitch grouping process is easy as position of each character is known. For typeset characters distance between characters are variable. The distance between words are significantly large than distance between characters and grouping is therefore possible. The Problems occur for handwritten characters when text is skewed. Until grouping each character is treated separately, the context in which each character appears has not been fully exploited. In our work to recognize characters, grouping is done by treating each character separately. As one writes characters in air at each own location, that particular location gets fixed for that line of characters. With increase in the characters location gets fixed for characters in each lines and grouping is done likewise.

4.3 APPLICATION OF OCR

The last few decades have seen a widespread appearance of commercial OCR products satisfying requirements of different users. In this section we highlight some notable application areas of OCR. The major application areas are often distinguished as data entry, text entry and process automation. The data entry area [19] covers technologies for entering large amounts of restricted data. Initially such machines were used for banking applications. The systems are characterized by reading only limited set of printed characters usually numerals and few special symbols. They are designed to read data like account numbers, customer’s identification, article numbers, amounts of money etc. The paper formats are constrained with a limited number of fixed lines to read per document. Because of these restrictions, readers of this kind may have a very high throughput up to 150 documents per hour. Due to limited character set these readers are usually tolerant to bad printing quality. These systems are specially designed for their applications and prices are therefore high.

The text entry reading machines [19] are used as page readers in office automation. Here the restrictions on paper format and character set are exchanged for constraints concerning font and printing quality. The reading machines are used to enter large amounts of text, often in word processing environment.

The above mentioned application areas are those in which OCR has been successful and widely used. However, many other areas of applications exist and some of which are [19, 32]:

a. **Aid for Blind:** In the early days before digital computers and requirement for input of large amounts of data emerged this was an imagined application area for reading machines. Along with speech synthesis system such reader enables blind to understand printed documents.
b. **Automatic Number Plate Readers:** A few systems for automatic reading of number plates of cars exist. As opposed to other OCR applications, input image is not natural bi-level image and must be captured by very fast camera. This creates special problems and difficulties although character set is limited and syntax restricted.
c. **Automatic Cartography:** The character recognition from maps presents special problems within character recognition. The symbols are intermixed with graphics, text is printed at different angles and characters are of several fonts or even handwritten.
d. **Form Readers:** Such systems are able to read specially designed forms. In such forms all irrelevant information to reading machine is printed in colour invisible to scanning device. The fields and boxes indicating where to enter text is printed in this invisible colour. The characters are in printed or hand written upper case letters or numerals in specified boxes. The instructions are often printed on form as how to write each character or numeral. The processing speed is dependent on amount of data on each form but may be few hundred forms per minute. The recognition rates are seldom given for such systems.
e. **Signature Verification and Identification:** This application is useful for banking environment. Such system establishes the identity of writer without attempting to read handwriting. The signature is simply considered as pattern which is matched with signatures stored in reference database.

CHAPTER 5

RESULT AND DESCRIPTION

5.1 DETECTION AND TRACKING

5.1.1 Initial Snapshot

The proposed method tracks the colored object in hand or a colored fingertip in a video file by capturing the video from any video-capturing device. Then it reads the video file and processes the video frame-wise by reading the images.First we have to start up the video stream using the command imaq.VideoDevice. Once the video stream is live we will continuously take snapshots and work on each one of them in the same manner. To take a snapshot the 'step' command is used. And it is ensured that the resolution is same as that of the laptop and size of each matrix returned is noted for further working on it.

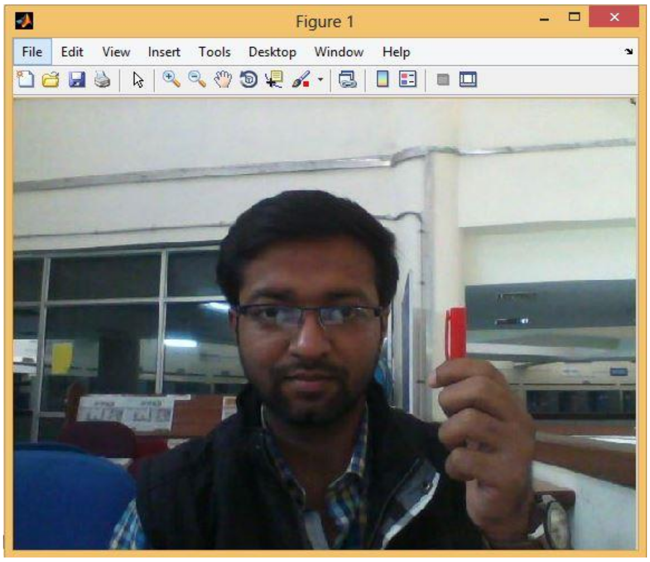


Fig. 5.1 A Snapshot of video stream

5.1.2 **Extracting Red Component**

Now to extract the red component part of the image we will convert the image into its grey form by using the command rgb2grey which finds out the grey component of a particular pixel by using 29.89% of the red component, 58.70% of the green component and 11.40% of the blue component. From this obtained image we subtract the red column of the image to get the red component image as a separate one. For extracting any other component other than red, that particular column can be subtracted, thus it is very easy to edit this program for any other color.

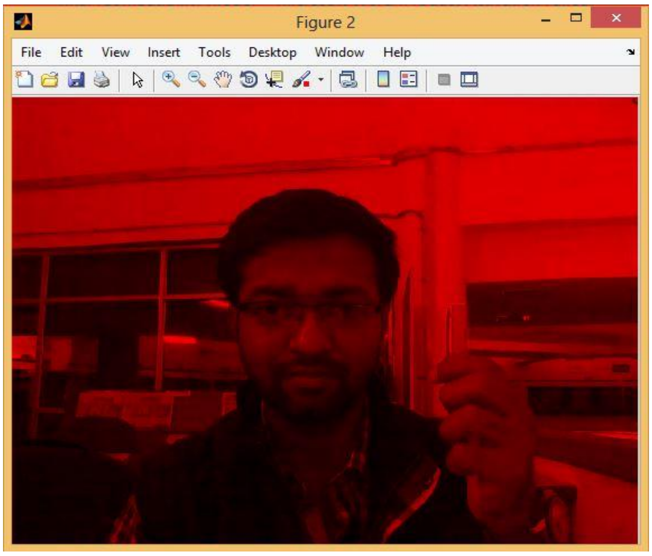


Fig. 5.2 Image containing Red Column only

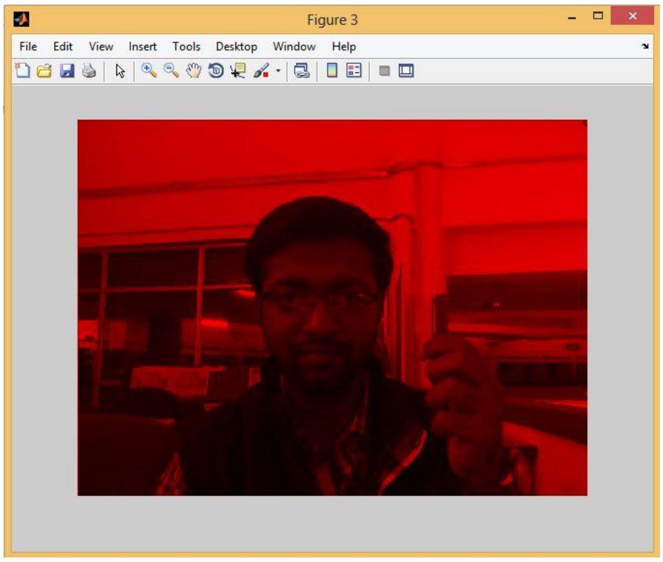


Fig. 5.3 Gray scale Image

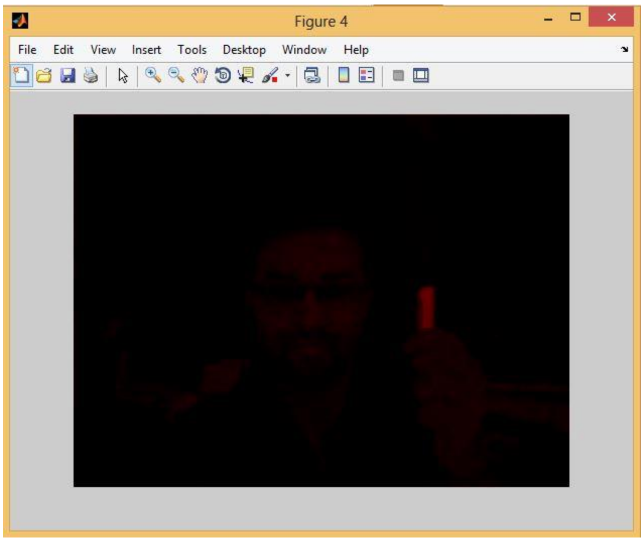


Fig. 5.4 Snapshot after subtraction (only red component)

5.1.3 **Filtering Noise**

Now we will filter noise using the median filter. Although other filters like the Gaussian and the mean filter could also be used, but the problem with those filters is that they do not provide edge enhancement as good as the median filter and plus they are less better at removing noise. We have already explained the working of the median filter in chapter 3 sixth section.

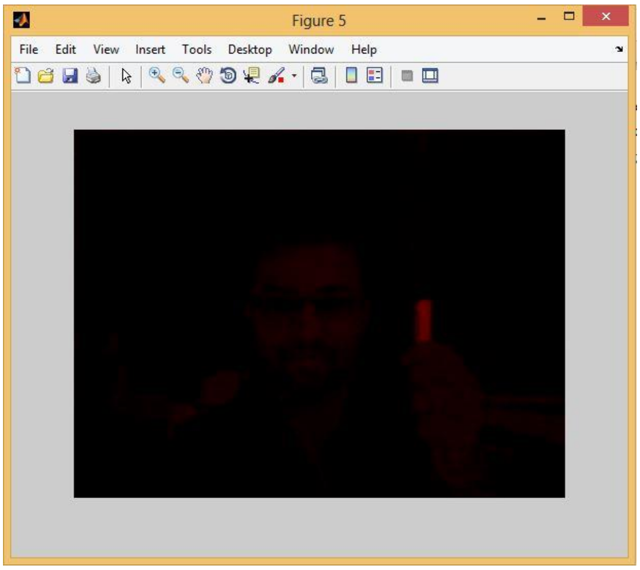


Fig. 5.5 Snapshot after using Median Filter

5.1.4 **Conversion of Snapshot to a Binary Image**

The final binary image is used to detect the red objects because of the fact that after proper experimentation we have defined a threshold value according to our surroundings and then we have only allowed values according to the threshold level to show in the binary image.

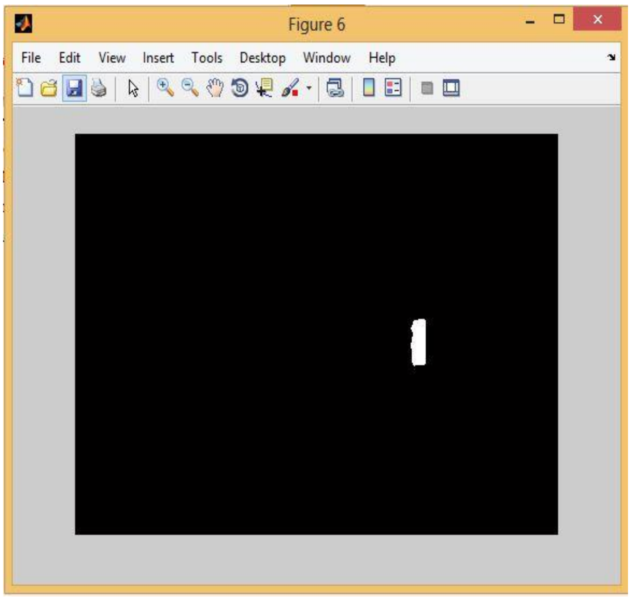


Fig. 5.6 Binary Image of Snapshot

We can see that it is very difficult to observe the red object through a naked eye but the important part is to notice that the object is of a significantly different color than the rest of the image thus making it clearly apparent thus helping us identify the object.

5.1.5 **Blob Recognition**

Finally we know which area is the one in which our object resides and after extracting the properties of the object from the still video frame, the object is tracked throughout the video frame using Blob Recognition method and its *(X, Y)* coordinates are obtained, as shown in Fig. 5.7. After the extraction of the *(X, Y)* coordinates, these coordinates are presented in the form of an image, which is further taken into consideration of future processing.

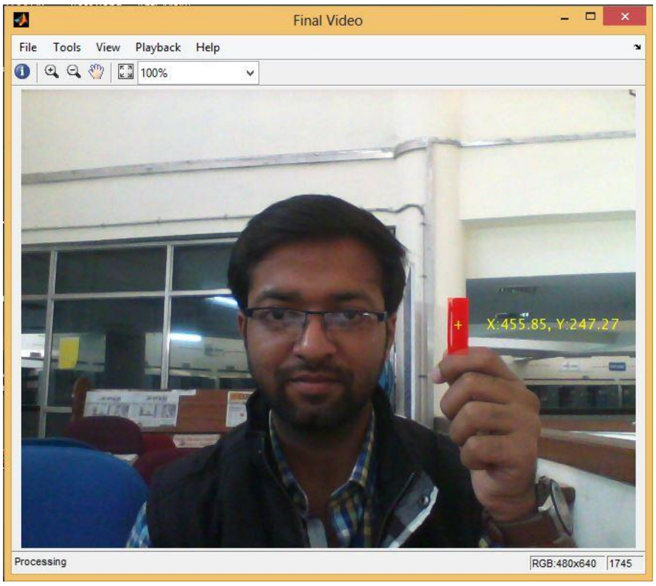


Fig. 5.7 Detected Object with bounding box using Blob recognition.

Here, the mirror images are obtained by simply shuffling the *X* coordinates rather than inverting the image, while the *Y* coordinates remain the same. Fig. 5.8 shows the plotted image of characters which get traced using Blob recognition method by continuously displaying the *(X, Y)* coordinates of the detected colored finger or any other colored object. Withdraw the colored object from the camera's vicinity or move to a non colored finger region to end drawing a character and then for re-drawing next character wait for some time (approximately 4 sec), then place the colored fingertip to the location where we want to start redrawing.

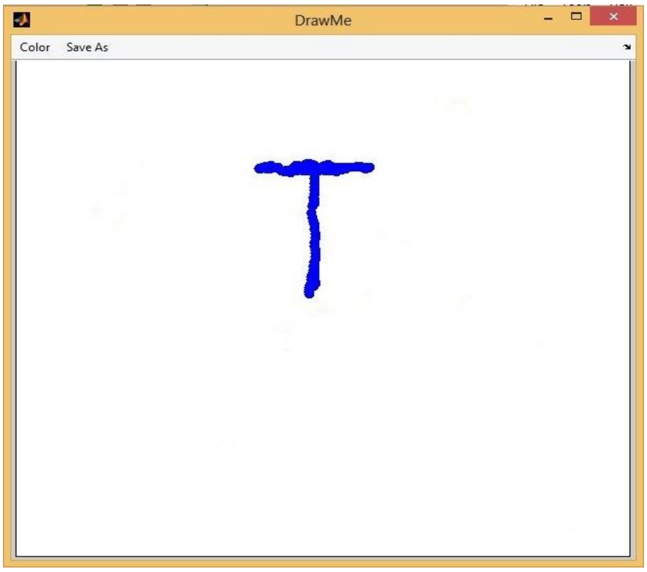


Fig. 5.8 Image of a Traced character.

5.2 CHARACTER RECOGNITION

5.2.1 Image-Acquisition

After every complete character traced when we remove the colored object/ finger from camera's vicinity a snapshot is taken using Matlab command 'getframe' and save that snapshot in an Image format such as .JPEG, .BMP etc. So after every characters is traced a snapshot of the whole frame is taken and OCR is applied for further processing as seen in Fig. 5.9.

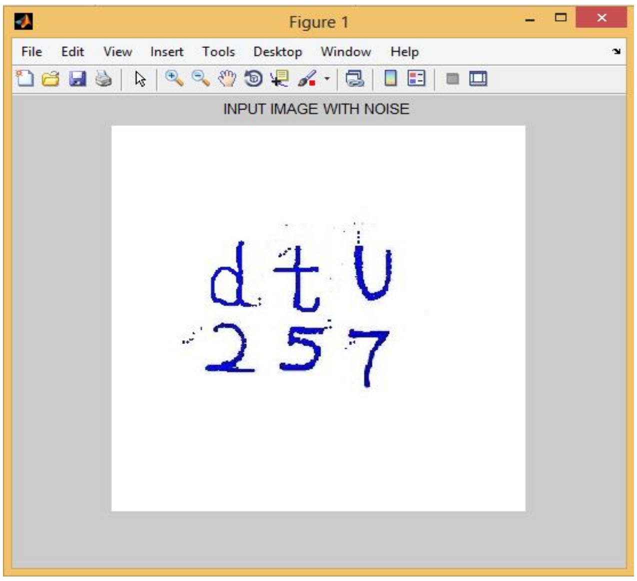


Fig. 5.9 Snapshot of Traced characters with Noise.

5.2.2 Pre-Processing

The captured Image of characters is further processed for subsequent OCR steps. Pre-processing steps includes first conversion of acquired image to binary image which include the intermediate steps of Grayscale conversion and then using threshold to the grayscale image which can be seen in the Fig. 5.10 and Fig. 5.11. As the captured image carries some shattered Noise portion due to the process of tracing the characters so, for cleaning the image morphological operation is applied on the binary converted image.

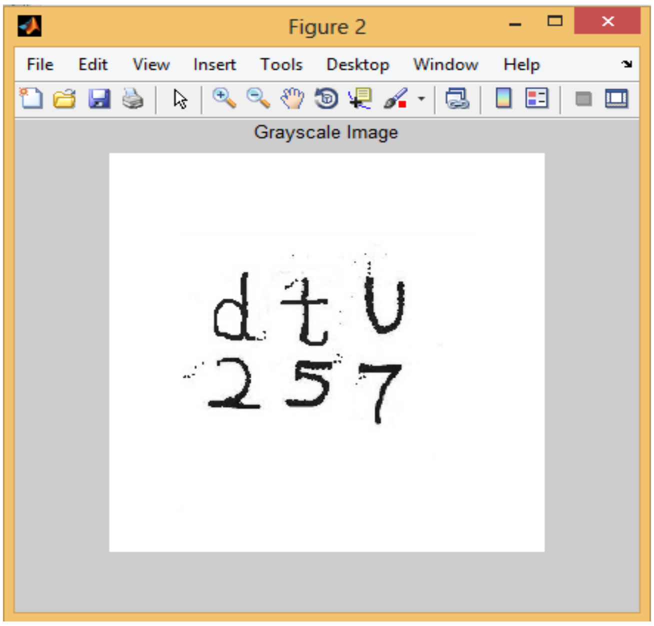


Fig. 5.10 Grayscale converted image of traced characters.

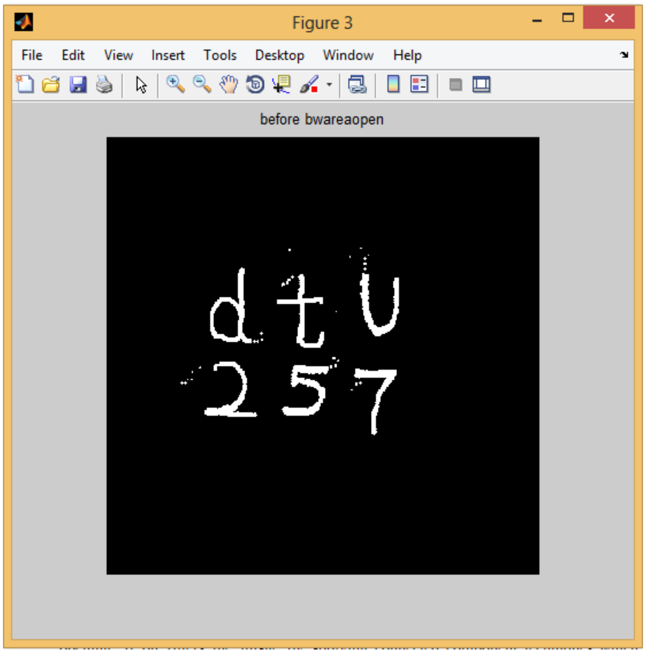


Fig. 5.11 Binary converted image of traced characters.

This is processed by using matlab command 'bwareaopen(I, P)', which removes from a binary image all connected components (objects) that have fewer than P pixels, producing another binary image BW2. This operation is known as an area opening. It on filters the image by applying connected component techniques which removes the objects containing fewer than 30 pixels (designated value), which removes the unnecessary objects called noise in an Image as can be seen in Fig. 5.12.

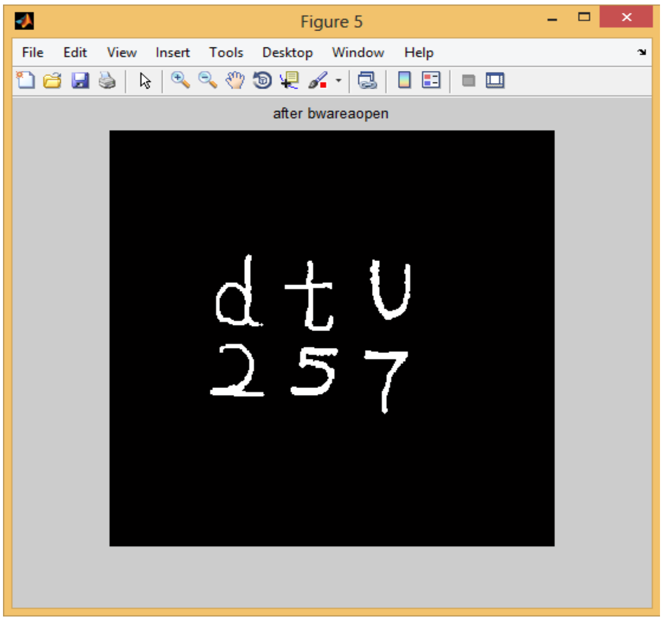


Fig. 5.12 Cleaned image after applying morphological operation (without Noise).

5.2.3 **Segmentation**

After the Image is cleaned and before the Template matching procedure is applied there is an intermediate step of segmenting the whole image with characters into segments of characters which can then be used for template matching process. The process of segmentation includes the image segmentation line-wise i.e. the whole image is firstly divided in to region of horizontal lines in the text written area i.e. clips the text written area into horizontal clippings of text areas as can be seen in the Fig. 5.13.

After Line-wise segmentation of the whole image, now each horizontal segmented line is further clipped into regions where single character exists using the connected component labelling. It is done with the help of Matlab command 'bwlabel (I, N)', which returns a matrix L, of the same size as an Image I, containing labels for the connected components in BW. N can have a value of either 4 or 8, where 4 specifies 4-connected objects and 8 specifies 8-connected objects; if the argument is omitted, it defaults to 8. The pixels labelled 0 are the background. The pixels labelled 1 make up one object, the pixels labelled 2 make up a second object, and so on. So, each labelled component is treated as a different region in which any alphanumerical number can be found. Now for applying the template matching algorithm each extracted characters needs to be converted to the size of template which is of size 42 x 24. Each extracted character is resized by using command ‘imresize’. Each resized extracted character can be seen from Fig. 5.14 to Fig. 5.19.

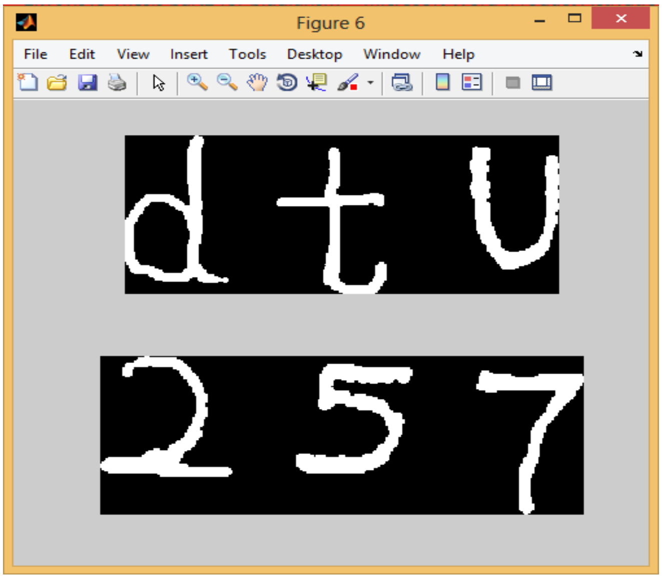


Fig. 5.13 Horizontal clipping of Cleaned Image using connected component technique.

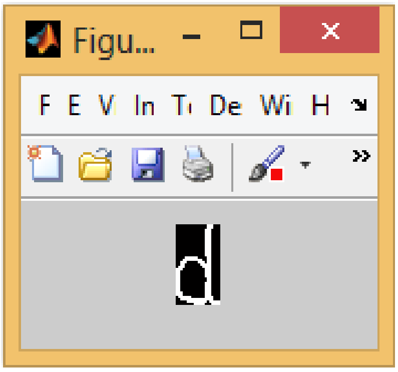


Fig. 5.14 Resized extracted character ‘D’

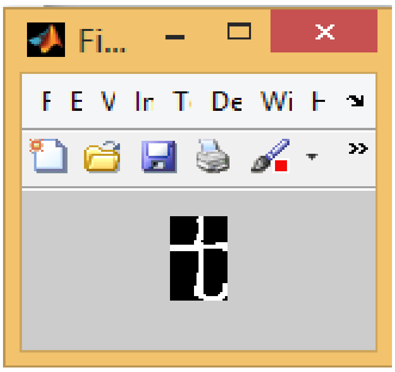


Fig. 5.15 Resized extracted character ‘T’.

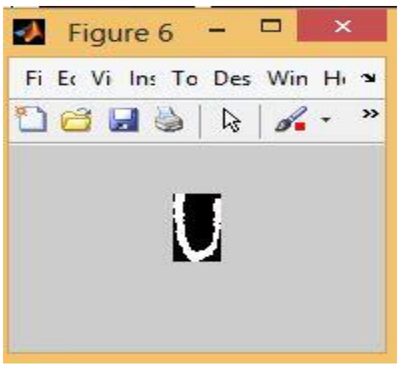


Fig.5.16 Resized extracted character ‘U’.

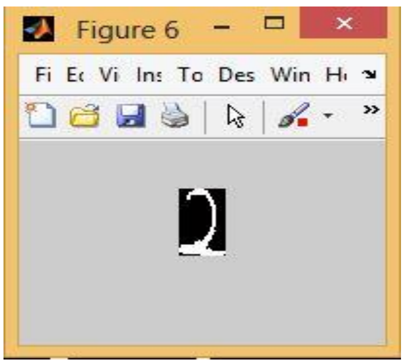


Fig. 5.17 Resized extracted numeral2’.

Fig. 5.18 Resized extracted numeral ‘5’.

Fig. 5.19 Resized extracted numeral ‘7’.

5.2.4 **Recognition of Text**

These resized images as seen from Fig. 5.14 to Fig. 5.19 are then given to the OCR module for recognition. It first loads the training test templates from the database as was seen in Fig. 4.3. After loading the complete template for alphanumeric characters, it is compared with the resultant image to find the maximum similarity as explained in chapter 4 section 2.4, and the result is displayed in a text file showing the resultant characters line-wise, as shown in Fig. 5.20 below.

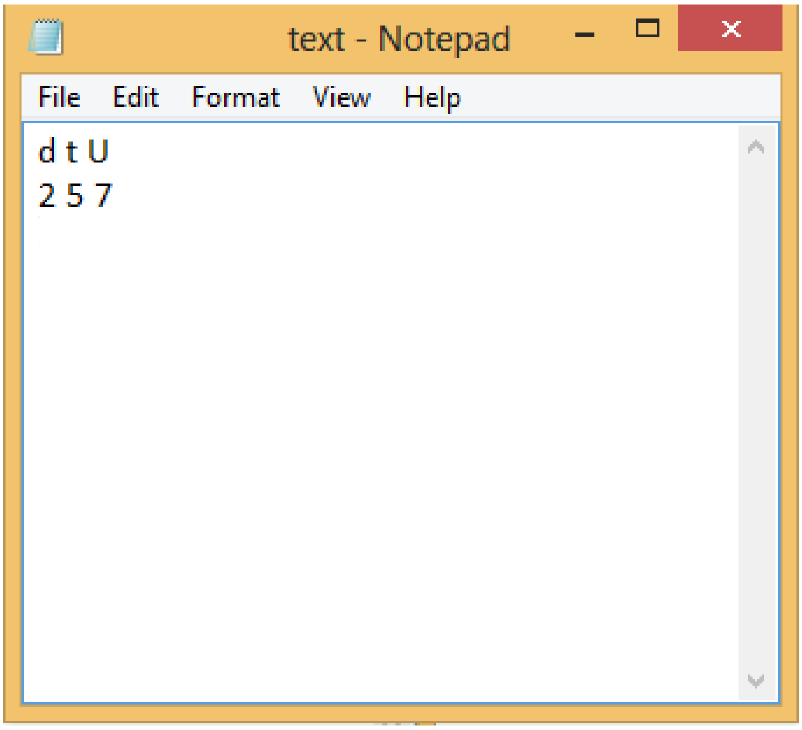


Fig. 5.20 Text file containing the recognized characters.

5.3 FLOWCHART AND PERFORMANCE OF PROPOSED ALGORITHM

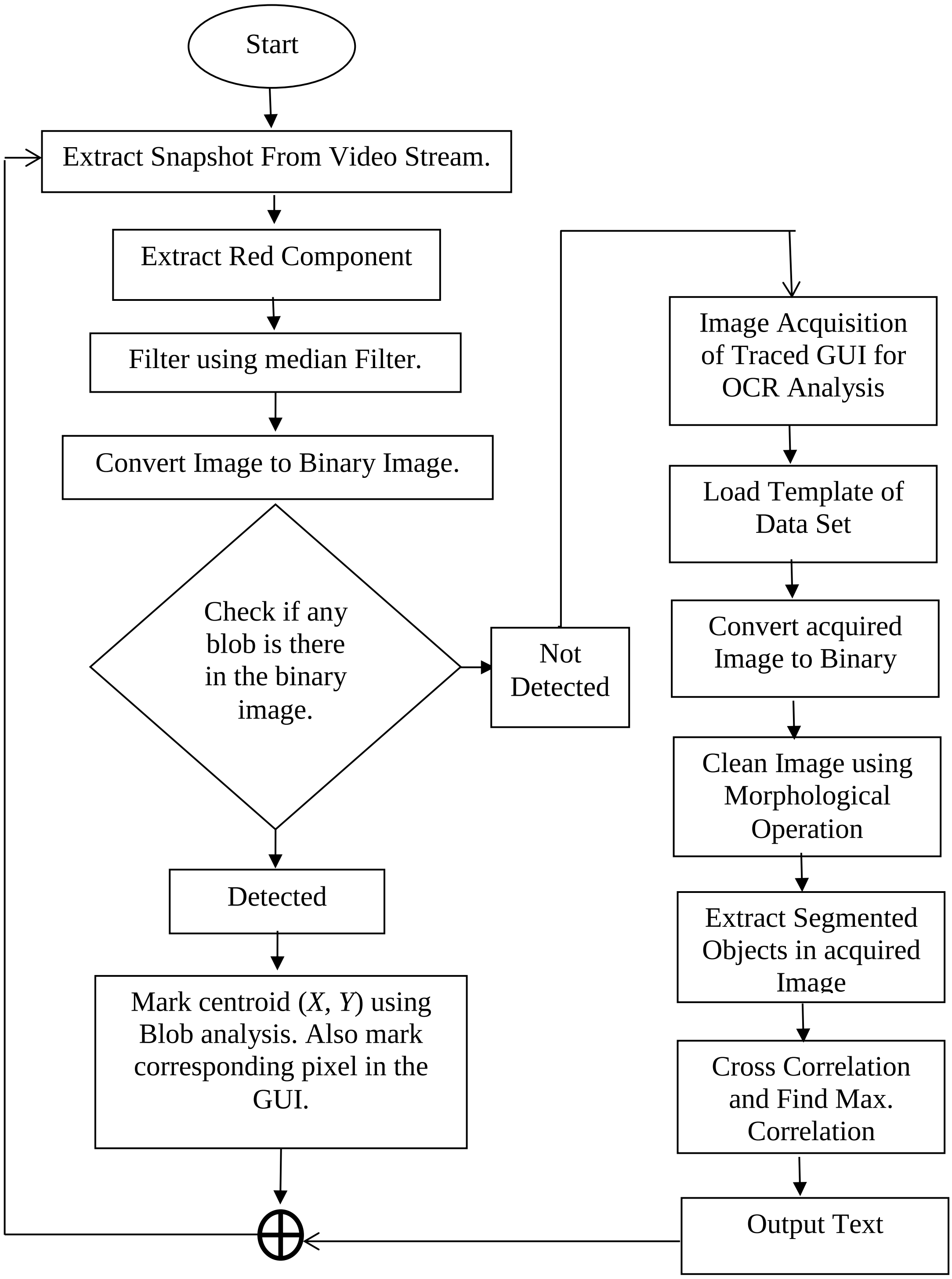
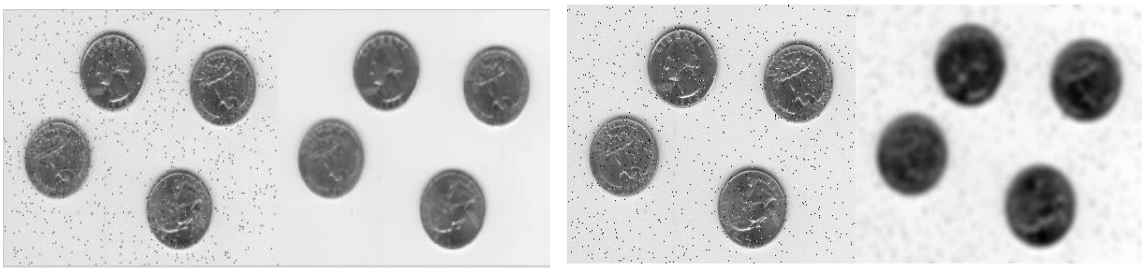


Fig. 5.21 Flowchart of the Proposed Algorithm

Each Step taken in our proposed algorithm is shown in Fig. 5.21 with the help of a Flowchart. In first phase of our work of extracting red color from the video, the first step is taking snapshots from the live video to process it for extracting and tracking red color. Red color is then extracted from the snapshot and then median filter is used for cleaning the image for better extraction of Red color. Reference [25] used Gaussian filter for smoothening the image in the process of extracting red color from the video frame. Median filter provides better result in smoothening the noisy image as can be seen in Fig. 5.22 thus making our algorithm more reliable in extracting the main red color portion from the image rather than some dispersed small pieces of red in background. After detecting our choice of object, centroid is marked using Blob analysis and rectangular grid is made covering the object where it is present in the frame. Same process it carried out on consecutive frames thus tracking the red object and making its trajectory on a GUI until it disappear from the Video frame for carrying out next part of character recognition of our work. After completion of tracking a character whenever our red object is deliberately pulled out from the video frame whole process gets paused for some 3sec to readjust the position to track new character. Meanwhile our process moves to the next part of recognizing that character whatever is being traced on the GUI.



(a) Filtered Image using Median filter (b) Filtered Image using Median filter
Fig. 5.22 Comparison of Median and Gaussian filter on a Noisy image.

Character recognition part is carried out initially by acquiring the frame of GUI traced for further processing it. Loading of saved Templates i.e. database of upper and lower case English alphabets (‘A-Z’ & ‘a-z’) and numerals (0-9) is carried out for future need in Template matching process. These templates are being saved in binary (1/0) format. Then the acquired image is passed through some pre-processing steps for cleaning the image. First step is to convert the acquired image to its binary equivalent for matching purpose by passing it through grayscale conversion step.

While drawing the characters one after another by removing our red object, placing it the next position of drawing and then make it appear again for drawing next character here comes the role of the processor of system and resolution of the camera which is being used. If our system is not high end then in this process of disappearing and again appearing our object at next desired location, some patch of the path gets faintly traced which make our traced image noisy as already seen in Fig. 5.9. For eliminating such unwanted noise we have make our process more robust by using morphological operation for removing objects containing fewer than 30 pixels, thus making our traced image free from unwanted traced objects as can be seen in Fig. 5.12. This step of unwanted noise removal makes our algorithm more robust and accurate than that was proposed by Faisal baig *et. al*.[25]

Robustness can be seen by evaluating performance measure of our proposed algorithm by collecting different samples of alphanumeric characters. Dataset is collected by freehand tracing the alphanumeric characters repeatedly for several times. Each upper and lower case characters ('A-Z' and 'a-z') and numerals ('0-9') are drawn iteratively for 40 times thus making a dataset of about 2520 samples. For instance dataset for performance evaluation of character 'A' can be seen in Fig. 5.23.

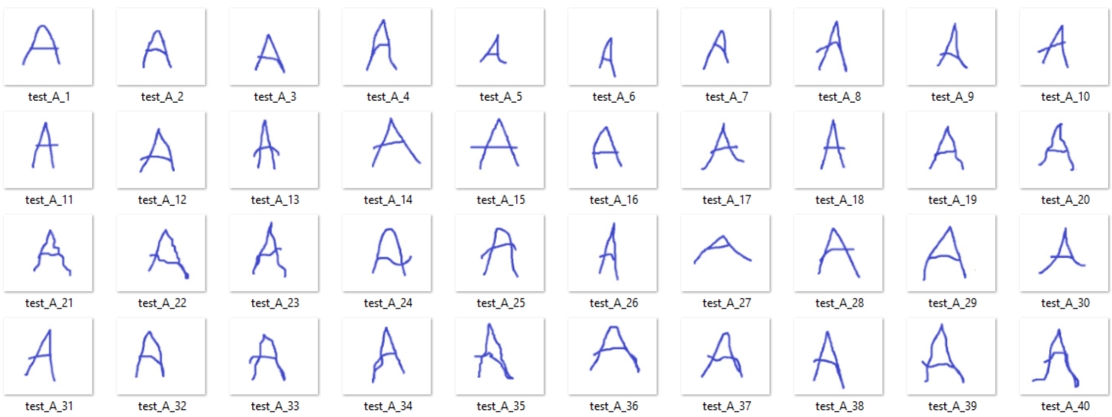


Fig. 5.23 Illustration of dataset for evaluating performance measure for character 'A'.

After every complete tracing and character recognition process, the count of correctly recognized samples is maintained for measuring the percentage accuracy of shown character ‘A’ in Fig. 5.23. Likewise this process is repeated for all alphanumeric samples and result for accurately detected samples in percentage are summarized in Table 5.1. After computing the complete accuracy our proposed algorithm was calculated to be 93.126 % thus improving the accuracy from the method proposed by Faisal Baig. *et al* which was 92.083%.

Table 5.1 Performance measure of character Recognition

Symbol	Accuracy (%)	Symbol	Accuracy (%)	Symbol	Accuracy (%)	Symbol	Accuracy (%)
‘A’	96	‘R’	96	‘i’	90	‘z’	96
‘B’	89	‘S’	96	‘j’	92	‘1’	89
‘C’	95	‘T’	91	‘k’	96	‘2’	96
‘D’	82	‘U’	87	‘l’	89	‘3’	100
‘E’	100	‘V’	83	‘m’	98	‘4’	100
‘F’	98	‘W’	97	‘n’	95	‘5’	89
‘G’	94	‘X’	95	‘o’	88	‘6’	95
‘H’	93	‘Y’	90	‘p’	94	‘7’	93
‘I’	90	‘Z’	95	‘q’	92	‘8’	88
‘J’	91	‘a’	95	‘r’	90	‘9’	94
‘K’	98	‘b’	94	‘s’	96	‘0’	85
‘L’	98	‘c’	97	‘t’	87	‘Space’	91
‘M’	100	‘d’	91	‘u’	97		
‘N’	100	‘e’	96	‘v’	98		
‘O’	84	‘f’	91	‘w’	92		
‘P’	98	‘g’	95	‘x’	96		
‘Q’	88	‘h’	89	‘y’	89		

CHAPTER 6

CONCLUSION

This dissertation presents a live video-based character recognition method that allows the writing of English text in the air using a device camera. The proposed method involves two main tasks: tracking the colored fingertip/ colored object in the video frames and then applying English OCR on the plotted images to recognize the written characters. It is relevantly very simple, fast, and easy to use and only require a camera and red tape or any color object holding in hand and a camera connecting to device on which it runs.

We learn that there a multiple problems whenever we are dealing with image processing. The lighting of the room is one big factor and we have to train our machine in such an efficient manner that it is able to overcome all of that. Our proposed method is applicable to all disconnected languages but it has one serious issue: due to color-sensitive, the existence of any red-colored object in the background before the start of and during the analysis can lead to false results. Also our proposed algorithm shows least accuracy for the letters D and V. The reason behind such a least accuracy rate is that while air-writing the letter D, it is sometimes apprehended in such a way that it looks similar to the letter O. A similar case is true with the letter V, which is sometimes captured as the letter U or W. As a user as we can simultaneously see the trajectory forming by little cautious on the edges of the character these issue can be minimized. The overall character recognition accuracy comes out to be 93.126%.

We also see that our project might find multiple applications in places that are in attention nowadays especially those parts which are growing. It is especially useful for such user interfaces that enables the user to save their without even typing on a keyboard or write on a track-pad/ touchscreen especially for handicapped persons, or for text input for smart system control, these are among many applications where this can be of good use.

REFERENCES

[1] Chen, Mingyu, Ghassan AlRegib, and Biing-Hwang Juang. "Air-writing recognition—Part I: Modeling and recognition of characters, words, and connecting motions." *IEEE Transactions on Human-Machine Systems* 46.3 (2016): 403-413.

[2] A. Yilmaz, O. Javed, M. Shah, ACM Computer Survey. 38, 1 (2006).

[3] Figueroa, Javier Poveda, and Vladímir Robles Bykbaev. "Image retrieval based on the combination of RGB and HSV's histograms and Colour Layout Descriptor." *Ingenius* 7 (2012): 3-10.

[4] Mohammad, Faisal, et al. "Optical character recognition implementation using pattern matching." *International Journal of Computer Science and Information Technologies* 5.2 (2014): 2088-90.

[5] Marial, A., and J. Jibrael. "Feature extraction of optical character recognition: survey." *Int. J. Appl. Eng. Res* 12.7 (2017): 1129-1137.

[6] Arica, Nafiz, and Fatos T. Yarman-Vural. "An overview of character recognition focused on off-line handwriting." *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 31.2 (2001): 216-233.

[7] S. Manke, M. Finke, A. Waibel, "NPen: A Writer Independent, Large Vocabulary On-Line Cursive Handwriting Recognition System", in *Proc. 3rd Int. Conf. Document Analysis and Recognition*, pp.403-408, Montreal, 1995.

[8] P. Carrieres, R. Plamondon, "An Interactive Handwriting Teaching Aid", in *Advances in Handwriting and Drawing : A Multidisciplinary Approach,* C. Faure, G. Lorette, A. Vinter, P. Keuss Eds., Paris, 1994, pp.207-239.

[9] R. Kashi, J. Hu, W. L. Nelson, W. Turin, "A Hidden Markov Model Approach To On-line Handwritten Signature Verification", *Int. Journal Document Analysis and Recognition*, vol.1, no.2, pp.102-109, 1998.

[10] S. N. Srihari, E. J. Keubert, "Integration of Handwritten Address Interpretation Technology into the United States Postal Service Remote Computer Reader System", in *Proc. 4th Int. Conf. Document Analysis and Recognition* , pp.892-896, Ulm, Germany, 1997.

[11] Knerr, Stefan, Valery Anisimov, Olivier Baret, Nikolai Gorski, David Price, and Jean-Claude Simon. "The A2iA intercheque system: courtesy amount and legal amount recognition for French checks." In *Automatic Bankcheck Processing*, pp. 43-86. 1997.

[12] S. Gopisetty, R. Lorie, J. Mao, M. Mohiuddin, A. Sorin, E. Yair, "Automated forms-processing Software and Services", *IBM Journal of Research and Development,* vol. 40, no. 2, pp.211-230, 1996.

[13] H. E. S. Said, T. N. Tan, K. D. Baker, "Personal Identification Based on Handwriting", *Pattern Recognition*, vol.33, no.1, pp.149-160, 2000.

[14] Q. Tian, P. Zhang, T. Alexander, Y. Kim. "Survey: Omnifont Printed Character Recognition", *Visual Communication and Image Processing 91: Image Processing*, pp. 260 - 268, 1991.

[15] I. Bazzi, R. Schwartz, J. Makhoul, "An Omnifont Open Vocabulary OCR System for English and Arabic", *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol.21, no.6, pp.495-504, 1999.

[16] H. I. Avi-Itzhak, T. A. Diep, H. Gartland, "High Accuracy Optical Character Recognition Using Neural Network with Centroid Dithering", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.17, no.2, pp.218-228, 1995.

[17] C. C. Tappert, "Adaptive on-line handwriting recognition", in *Proc. 7th Int. Conf. Pattern Recognition*, Montreal, Canada, pp. 1004-1007, 1984.

[18] Srihari S N (1992) High-Performance Reading Machines. Proceedings of the IEEE, 80(7): 1120-32.

[19] Nagy, George, Thomas A. Nartker, and Stephen V. Rice. "Optical Character recognition: An illustrated guide to the frontier." *Document Recognition and Retrieval VII*. Vol. 3967. International Society for Optics and Photonics, 1999.

[20] Keysers, Daniel, et al. "Multi-language online handwriting recognition." *IEEE transactions on pattern analysis and machine intelligence* 39.6 (2017): 1180-1194.

[21] Hannuksela, Jari, et al. "Adaptive motion-based gesture recognition interface for mobile phones." *International Conference on Computer Vision Systems*. Springer, Berlin, Heidelberg, pp. 271-280, May 2008.

[22] Asano, Toshio, and Sachio Honda. "Visual interface system by character handwriting gestures in the air." *RO-MAN, 2010 IEEE*. IEEE, pp. 56-61, 2010.

[23] Vikram, Sharad, Lei Li, and Stuart Russell. "Handwriting and Gestures in the Air, Recognizing on the Fly." *Proceedings of the CHI*. Vol. 13, pp. 1179-1184, 2013.

[24] Mohammad, Faisal, et al. "Optical character recognition implementation using pattern matching." *International Journal of Computer Science and Information Technologies* 5.2, pp. 2088-2090, 2014.

[25] Baig, Faisal, Muhammad Fahad Khan, and Saira Beg. "Text writing in the air." *Journal of Information Display* 14.4, pp. 137-148, 2013.

[26] M. Ziaratban, K. Faez, F. Faradji, “Language-based feature extraction using template-matching in Farsi/Arabic handwritten numeral recognition”, Ninth International Conference on Document Analysis and Recognition, pp. 297 - 301, 2007.

[27] Alper Yilmaz, Omar Javed, and Mubarak Shah, “Object tracking: A survey.” *Acm Computing Surveys (CSUR),* 38(4):13, 2006.

[28] Singh, Pritpal, et al. "Real-time object detection and tracking using color feature and motion." *Communications and Signal Processing (ICCSP), 2015 International Conference on*. pp. 1236-1241, 2015.

[29] Varshney, Surabhi, Rashmi Chaurasiya, and Yogesh Tayal. "Optical Character Recognition using Neural Network." *International Journal of Emerging Technology and Advanced Engineering* 4.8 (2014): 711-715.

[30] Rafael C. Gonzalez, Richard E. woods and Steven L.Eddins, Digital Image Processing using MATLAB, Pearson Education, Dorling Kindersley, South Asia, 2004.

[31] Chaudhuri, A., Some Experiments on Optical Character Recognition Systems for different Languages using Soft Computing Techniques, Technical Report, Birla Institute of Technology Mesra, Patna Campus, India, 2010.

[32] Raghavendra S P, Ajit Danti and Suresha .M, “Correlation Based Template Matching For Recognition Of Bank Cheque Number”, International Journal of Computer Engineering and Applications, Volume XII, Issu, pp. 166-181, January 2018.

[33] Cheriet, M., Kharma, N., Liu, C. L., Suen, C. Y., Character Recognition Systems: A Guide for Students and Practitioners, John Wiley and Sons, 2007.